On the adequacy of per models

Roberto M. Amadio

UNITÉ DE RECHERCHE
INRIA-LORRAINE

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P. 105
78153 Le Chesnay Cedex
France
Tél.: (1) 39 63 55 11

Rapports de Recherche

N° 1579

Programme 2
Calcul Symbolique, Programmation et Génie logiciel

ON THE ADEQUACY OF PER MODELS

Roberto M. AMADIO

Janvier 1992

RR-1579

On the Adequacy of Per Models

Roberto M. Amadio
CRIN-CNRS, Nancy[1]

Abstract

In this note we consider a fixed point extension of the second order lambda calculus equipped with a call by value evaluation mechanism. We interpret the language in a partial cartesian closed category of "directed complete" partial equivalence relations (pers) over a domain theoretic model of a type-free, call-by-value, lambda calculus. Our main result is that the notions of "syntactic" and "semantic" convergence coincide.

Résumé

Dans cette note nous considérons une extension de point fixe du lambda calcul du deuxième ordre avec un mécanisme d'évaluation par valeur. Nous interprétons ce calcul dans une catégorie cartésienne partialement fermée de relations partiels d'équivalence. Le résultat principal est l'équivalence des notions syntaxique et sémantique de convergence.

[1] Group "Prograis", URA 262, CRIN-CNRS, B.P. 239, 54506, Vandoeuvre-lès-Nancy, FRANCE. E-mail: amadio@loria.crin.fr. "Prograis" is also a joint project with INRIA-Lorraine.

Introduction

Recently the research in the area of "synthetic domain theory" (see, e.g., Hyland[91]) has especially addressed the problem of discovering subcategories of partial equivalence relations (pers) over a partial combinatory algebra (pca) that enjoy good completeness properties, and that admit certain constructions typical of domain theory, such as the solution of recursive function and domain equations.

Our concern here does not lie in the construction of models, or in the categorical abstraction of such construction, but in an attempt at connecting such models to issues arising in the design and semantics of, say, typed functional languages. In particular we connect a certain per-interpretation to a call-by-value evaluation discipline that corresponds to current implementations of higher order typed functional languages with a static type checker (see, e.g., Cardelli[89]).

The *main result* establishes the equivalence of the syntactic and semantic notions of convergence. This is a classical "adequacy" result for domain theoretic interpretations, as sketched for example in Plotkin[85], after Martin-Löf[83]. However, as far as we know, no results were available, in the case of per-interpretations.

There are two additional points we wish to emphasize:

• In order to prove such an adequacy theorem *very little structure* on the per model is needed, in particular we do not require working with an O-category.

• The adequacy of the per interpretation is largely *independent* from the adequacy of the underlying pca.

Section 5 is the technical core of this paper. The proof of adequacy w.r.t. standard domain theoretic interpretations, requires the combination of "admissible predicates" techniques and "reducibility candidates" techniques. In the proof we propose here, there are two additional twists that are due to the presence of second order types, and to the interplay between the typed and the type-free structures. In particular the key of the result lies in the definition of *adequacy relation* (5.1), and *in the way one associates an adequacy relation to a type* (5.2).

In order to make the central result quickly accessible we have delayed the more or less standard proofs of the first four sections to appendix A. Such sections contain respectively: (1) the definition of a fixed point extension of the second order lambda calculus; (2) the definition of the evaluation mechanism of such language; (3) the description of the basic properties of the partial cartesian closed category of "directed complete" pers over a cpo model of call-by-value, type-free lambda calculus; (4) the interpretation of the language in the semantic structure.

1. Language

Types and raw terms are defined by the following BNFs:

Type Variables:	$tv ::= t \mid s \mid \ldots$
Types:	$\alpha ::= 1 \mid tv \mid (\alpha \rightarrow \alpha) \mid (\forall tv.\alpha)$
Term Variables:	$v ::= x \mid y \mid \ldots$
Terms:	$M ::= * \mid v \mid (\lambda v{:}\alpha.M) \mid (MM) \mid (\lambda tv.M) \mid (M\alpha) \mid (Y_\alpha M)$

In the following we will feel free to spare on parentheses, and to omit the type label in the Y combinator. All types and terms are considered up to α-redenomination.

A *well formed context* Γ is given by a list of pairs, v: α, in which all variables are distinct. We write Γ, v: α to evidentiate the last element of the list, we write $v{:}\alpha\in\Gamma$ to state that $v{:}\alpha$ occurs in Γ, and we denote with ftv(Γ) the collection of type variables free in types occurring in Γ. Note that in the calculus presented here type variables contexts are left implicit.

As usual a *substitution*, say σ, is a function that associates variables, say v, v_1, ..., to formal expressions, say exp, exp_1, ..., . The domain of a substitution is $\{v \mid \sigma(v)\neq v\}$. We assume that such domain is always finite. We denote with $[exp_1/v_1,..., exp_n/v_n]$ a substitution whose domain is contained in $\{v_1,..., v_n\}$, and that associates exp_i to v_i, for i=1,...,n. If σ is a substitution and exp is an expression then σexp denotes the expression resulting from the application of the substitution to the expression, according to the standard rules that take care of bounded variables. We abbreviate an iteration of substitutions, say $\sigma_1(..(\sigma_n\ exp)..)$, with $\sigma_1..\sigma_n$ exp, so, for example, $[r/s][s/t]\,(t\rightharpoonup s) = (r\rightharpoonup r)$.

In a formal system the symbol "$\Rightarrow$" separates, in an inference rule, the premisses from the conclusion. If "J" is a judgment of the formal system then we write "$\vdash$J", if such judgment is derivable.

A *typing judgment* is of the shape $\Gamma\supset M{:}\alpha$ where Γ is always a well formed context. Derivable typing judgment are specified by the following formal system:

(*)	$\Rightarrow \Gamma \supset *: 1$
(asmp)	$x{:}\ \alpha\in\Gamma \Rightarrow \Gamma\supset x{:}\ \alpha$
($\rightharpoonup$I)	$\Gamma,x{:}\alpha \supset M{:}\beta \Rightarrow \Gamma\supset (\lambda x{:}\alpha.M){:}\ (\alpha\rightharpoonup\beta)$
($\rightharpoonup$E)	$\Gamma\supset M{:}\ (\alpha\rightharpoonup\beta)\ ,\ \Gamma\supset N{:}\alpha \Rightarrow \Gamma\supset(MN){:}\ \beta$
($\forall$I)	$\Gamma\supset M{:}\ \alpha\ \ t\notin ftv(\Gamma) \Rightarrow \Gamma\supset (\lambda t.M){:}\ (\forall t.\alpha)$
($\forall$E)	$\Gamma\supset M{:}\ (\forall t.\alpha) \Rightarrow \Gamma\supset (M\beta){:}\ [\beta/t]\alpha$
(Y)	$\Gamma\supset M{:}\ (1\rightharpoonup\alpha)\rightharpoonup\alpha \Rightarrow \Gamma\supset Y_\alpha M{:}\ \alpha$

This language is intended to represent a second order lambda calculus with a fixed point combinator over lifted types (one can think of the constant Y_α as having type $((\alpha)_\perp\rightarrow(\alpha)_\perp)\rightarrow(\alpha)_\perp$, where: $(\alpha)_\perp \equiv (1\rightharpoonup\alpha)$, and $(\alpha\rightharpoonup\beta) \equiv (\alpha\rightarrow(\beta)_\perp)$. The type $(\alpha\rightharpoonup\beta)$ should be thought as the type of the *partial* functions from α to β.

2. Evaluation

The *canonical forms* are the well-typed terms without free variables, but possibly with free type variables, that are generated by the following grammar:

$$C ::= * \mid (\lambda v{:}\ \alpha.\ M) \mid (\lambda tv.\ C)$$

The evaluation "$\mapsto$" is specified as a relation between terms without free variables and canonical forms. If $M\mapsto C$ then M and C have the same type, so one may also think of "$\mapsto$" as a family of relations indexed over types. The definition of the evaluation relation proceeds by induction on the structure of a well-typed closed term.

(*)	$\Rightarrow \; * \mapsto *$	
(asmp)	"we never evaluate a free variable"	
($\rightharpoonup$I)	$\Rightarrow \; \lambda x{:}\alpha.M \mapsto \lambda x{:}\alpha.M$	
($\rightharpoonup$E)	$M \mapsto \lambda x{:}\alpha.M',\; N \mapsto C',\; [C'/x]M' \mapsto C \;\Rightarrow\; MN \mapsto C$	
($\forall$I)	$M \mapsto C \;\Rightarrow\; \lambda t.M \mapsto \lambda t.C$	
($\forall$E)	$M \mapsto \lambda t.C \;\Rightarrow\; M\alpha \mapsto [\alpha/t]C$	
(Y)	$M(\lambda x{:}1.YM) \mapsto C \;\Rightarrow\; YM \mapsto C$	(for x fresh variable)

We write $M\downarrow$ if $\vdash M \mapsto C$, for some canonical form C, and $M\uparrow$ otherwise. Note that the definition of "$\mapsto$" gives directly a deterministic procedure to reduce, if possible, a closed term to a canonical form. Hence each term can reduce at most to a canonical form. Canonical forms always reduce to themselves.

Observe that we *evaluate under type abstraction*. On one hand this corresponds to the fact that in actual implementations of the language the type-checker is static, hence no information about type abstraction and type application appears at run time. On the other hand, as it will become clear in section 4, this choice is important in capturing the behavior of the interpretation of second order types as intersections.

3. Semantic Structure

In the presentation of the per-model we take a minimalist approach, by presenting only those properties and constructions that are needed in the proof. We refer to Amadio[90] for more information about the relevance and the context of the structures discussed below.

Conventions *(category of dcpos)*

A set X is *directed* in the poset $(P, \leq)$ if $\emptyset \neq X \sqsubseteq P$, and $\forall x,y \in X. \exists z \in X.(x \leq z \;\wedge\; y \leq z)$. A poset is *directed complete* (dcpo) if it has joins of its directed subsets. Two mathematical expressions including partial operations, say e_1, e_2, are *Kleene equivalent*, written $e_1 \simeq e_2$, if either they are both defined and they are equal, or they are both undefined. We also write $e\Downarrow$ ($e\Uparrow$) if a mathematical expression is defined (undefined). A *partial (Scott-)continuous function* between two dcpos, $h:(D,\leq_D) \rightharpoonup (E, \leq_E)$, is a partial function between the dcpos D and E such that for any directed set X in D, $f(\sqcup X) \simeq \sqcup f(X)$ (whenever we take the join of an indexed set of mathematical expressions, such join is defined if the join of the *defined* mathematical expressions is defined). We denote with **dcpo** the category of dcpos and partial continuous functions. This category is (equivalent to) a *partial cartesian closed category* in the sense of Moggi[88]. Given two dcpos, D, E, we denote with $D \rightharpoonup E$, the partial exponent, that is the collection of partial continuous functions pointwise ordered.

3.1 Realizability Structure

We assume to have an object D in the category of **dcpo** that has its partial functional space as a retract, i.e. i: $(D\rightharpoonup D)\rightharpoonup D$, j: $D\rightharpoonup(D\rightharpoonup D)$, $j\circ i = id_{D\rightharpoonup D}$ in **dcpo**. We define a partial operation of application over D as: $de \triangleq j(d)(e)$. From this operation one can define, as usual, continuous operations of pairing, $<\ ,\ >:D\times D\rightarrow D$, and projection $\pi_i:D\rightarrow D$, $(i=1,2)$ such that $\pi_1<d,d'> = d$, and $\pi_2<d,d'> = d'$.

More Conventions *(category of ppers)*

A partial equivalence relation over D (per) is a binary relation over D that is symmetric and transitive. We denote with A, B,... pers over D. We write: dAe for $(d,e)\in A$, $[d]_A$ for $\{e\in D \mid dAe\}$, $|A|$ for $\{d\in D \mid dAd\}$, $[A]$ for $\{[d]_A \mid d\in |A|\}$. A partial morphism between pers, f: $A\rightharpoonup B$, is a map f: $[A]\rightharpoonup[B]$ such that

$$\exists h\in D.\ \forall d\in |A|.\ (\ f([d]_A)\Downarrow\ \wedge\ hd\in f([d]_A))\ \vee\ (f([d]_A)\Uparrow\ \wedge\ hd\Uparrow)).$$

We denote with **pper** the category of pers and partial morphisms of pers. Such category is (equivalent to) a pccc where terminal, product (in the related category of total morphisms), and partial exponent pers are defined as follows:

$$\mathbf{1} \triangleq D\times D, \qquad dA\times Be \Leftrightarrow \pi_1 dA\pi_1 e \wedge \pi_2 dB\pi_2 e,$$
$$h\ \mathrm{pexp}(A, B)\ k \Leftrightarrow \forall d,e.\ dAe \Rightarrow (hdBke \vee (hd\Uparrow \wedge ke\Uparrow)).$$

The interpretation of the language is based on the following category of directed complete pers and partial maps. One may think of this category as a loose analogous of the category of dcpos and partial continuous maps. We will see that it retains the basic properties of the category of ppers, and moreover it has a fixed point combinator over "lifted" objects. The proofs of these results follow Amadio[89].

3.2 Definition *(directed complete pers)*

A per A is directed complete (dcper) if for any directed set X in $D\times D$, if $X\subseteq A$ then $\sqcup_{D\times D}X\in A$. We define **dcpper** as the full subcategory of **pper** whose objects are dcpers.

3.3 Proposition *(basic properties of the semantic structures)*

(1) **dcpper** is a pccc.

(2) dcpers are closed under arbitrary intersections.

(3) **dcpper** is reflective in **pper**, that is the inclusion functor from **dcpper** to **pper** has a left adjoint.

(4) **dcpper** has fixpoints over objects of the shape pexp(**1**,A).

4. Interpretation

In this section we define an interpretation of the language based on the semantic structures just introduced. By convention if $\tau\colon V \rightharpoonup W$ is a partial function from a collection of variables, v, to a set of values, W, then for $v \in V$, and $a \in W$ we define: $\tau[a/v](v') \simeq$ if $v' \equiv v$ then a else $\tau(v')$.

Types. The interpretation of a type, given a type assignment η: tv$\to$dcper, is a dcper, defined by induction as follows:

$$[\![1]\!]\eta = \mathbf{1}$$
$$[\![t]\!]\eta = \eta(t)$$
$$[\![\alpha \rightharpoonup \beta]\!]\eta = \mathrm{pexp}([\![\alpha]\!]\eta, [\![\beta]\!]\eta)$$
$$[\![\forall t.\alpha]\!]\eta = \bigcap_{A\ \mathrm{dcper}} [\![\alpha]\!]\eta[A/t]$$

Terms. An assignment is a partial function $\rho\colon v \rightharpoonup \bigcup_{A\ \mathrm{dcper}}[A]$. A type assignment η is compatible with an assignment ρ, w.r.t. a well-formed context Γ, if for any $x{:}\alpha \in \Gamma$, $([\![\alpha]\!]\eta = \varnothing \Rightarrow \rho(x)\Uparrow) \wedge ([\![\alpha]\!]\eta \neq \varnothing \Rightarrow \rho(x) \in [[\![\alpha]\!]\eta])$; we shortly write this as $\eta \uparrow_\Gamma \rho$. The interpretation of a judgment $\vdash \Gamma \supset M\colon \alpha$, given η, ρ, such that $\eta \uparrow_\Gamma \rho$, is either undefined or an element in $[[\![\alpha]\!]\eta]$ (equivalently it is a partial map from the terminal object to $[\![\alpha]\!]\eta$). Such interpretation is defined by induction on the length of the typing judgment. Observe that some clause may fail to be defined, hence the use of Kleene equivalence. Suppose $\vdash \varnothing \supset M\colon \alpha$, we write $M\Downarrow$ if for any type assignment η, and any ρ, $[\![\varnothing \supset M{:}\alpha]\!]\eta\rho \Downarrow$, and $M\Uparrow$ otherwise.

(*) $[\![\Gamma \supset *\colon 1]\!]\eta\rho = [d]_1$, for $d \in D$.

(asmp) $[\![\Gamma \supset x_i\colon \alpha_i]\!]\eta\rho \simeq \rho(x_i)$

($\rightharpoonup$I) $[\![\Gamma \supset \lambda x{:}\alpha.M{:}\alpha \rightharpoonup \beta]\!]\eta\rho = \{h \in D \mid \forall d \in |A|.\ (f(d)\Downarrow \Rightarrow hd \in f(d)) \wedge (f(d)\Uparrow \Rightarrow hd\Uparrow)\}$
where: $A = [\![\alpha]\!]\eta$, $f(d) \simeq [\![\Gamma, x{:}\alpha \supset M{:}\beta]\!]\eta\rho[[d]_A/x]$.

($\rightharpoonup$E) $[\![\Gamma \supset MN{:}\beta]\!]\eta\rho \simeq [\![\Gamma \supset M{:}\alpha \rightharpoonup \beta]\!]\eta\rho \cdot [\![\Gamma \supset N{:}\alpha]\!]\eta\rho$
where: $A \triangleq [\![\alpha]\!]\eta$, $B = [\![\beta]\!]\eta$, $[h]_{\mathrm{pexp}(A,B)} \cdot [d]_A \simeq [hd]_B$.

($\forall$I) $[\![\Gamma \supset \lambda t.M{:}\forall t.\alpha]\!]\eta\rho \simeq$ *if* $\exists A.(f(A)\Uparrow)$ *then* $\Uparrow$ *else* $\{h \in D \mid \forall A$ dcper. $h \in f(A)\}$
where: $f(A) \simeq [\![\Gamma \supset M{:}\alpha]\!]\eta[A/t]\rho$.

($\forall$E) $[\![\Gamma \supset M\beta{:}[\beta/t]\alpha]\!]\eta\rho \simeq [\![\Gamma \supset M{:}\forall t.\alpha]\!]\eta\rho \cdot [\![\beta]\!]\eta$
where: $F = \lambda A.[\![\alpha]\!]\eta[A/t]$, $\bigcap F = \bigcap_{A\ \mathrm{dcper}} F(A)$, $[h]_{\bigcap F} \cdot B \simeq [h]_{F(B)}$.

(Y) $[\![\Gamma \supset YM{:}\alpha]\!]\eta\rho \simeq [\ \bigsqcup_{n<\omega} k(n)\ ]_A$
where: $B = [\![(1 \rightharpoonup \alpha) \rightharpoonup \alpha]\!]\eta$, $[k]_B \simeq [\![\Gamma \supset M{:}(1 \rightharpoonup \alpha) \rightharpoonup \alpha]\!]\eta\rho$,
$k(0) \simeq \Uparrow$, $k(n+1) \simeq k\ i(\lambda d \in D.k(n))$.

Note: We retain the attention of the reader on three points:

(1) Something has to be done in order to show in the clauses ($\rightharpoonup$I) and ($\forall$I) that certain collections of realizers are not empty.

(2) When we apply a term to a type (clause ($\forall$E)), we keep the same realizer, this connects with the choice of evaluating under type abstraction.

(3) In the (Y) clause the existence of the fixed point combinator, which was announced in proposition 3.3.(4), takes a concrete shape. Its construction takes advantage of an iterator one can build in the realizability structure.

The following is proved by connecting the interpretation of a typed term in the per-model to the interpretation of its underlying type free-term in the realizability structure.

4.1 Proposition *(Typing Soundness)*

If $\vdash \Gamma \supset M{:}\alpha$ then, for any η and ρ such that $\eta \uparrow_\Gamma \rho$, we have:

$$[\![\Gamma \supset M{:}\alpha]\!]\eta\rho \Downarrow \ \Rightarrow\ [\![\Gamma \supset M{:}\alpha]\!]\eta\rho \in [\, [\![\alpha]\!]\eta\,].$$

Proviso *(on type substitution)*

In the following $\sigma, \tau, \ldots,$ denote substitutions of types for type variables. We now specify top-down what it means to apply such substitutions to a typing judgment. As usual one has to treat cautiously bound variables.

Judgment: $\sigma(\Gamma \supset M{:}\alpha) = \sigma\Gamma \supset \sigma M{:}\sigma\alpha$

Types: $\sigma 1 = 1;\ \sigma t = \sigma(t);\ \sigma(\alpha \rightharpoonup \beta) = \sigma\alpha \rightharpoonup \sigma\beta;$
$\sigma(\forall t.\alpha) = \forall r.\sigma[r/t]\alpha$, where r is a fresh variable.

Contexts: $\sigma(x_1{:}\alpha_1,\ldots, x_n{:}\alpha_n) = x_1{:}\sigma\alpha_1,\ldots, x_n{:}\sigma\alpha_n$

Terms: $\sigma * = *;\ \sigma x = x;\ \sigma(\lambda x{:}\alpha.M) = (\lambda x{:}\sigma\alpha.\sigma M);\ \sigma(MN) = \sigma(M)\,\sigma(N);$
$\sigma(\lambda t.M) = \lambda r.\sigma[r/t]M$, where r is a fresh variable; $\sigma(M\alpha) = \sigma M\sigma\alpha;$
$\sigma(Y_\alpha M) = Y_{\sigma\alpha}\sigma M$

Having defined the notion of type substitution in a typing judgment the next thing to verify is that provability is invariant under type substitution, and that type and term substitutions commute with the respective semantic substitutions. The following lemmas are proved by induction on the length of the typing.

4.2 Lemma *(Type Substitution)*

Suppose $\vdash \Gamma \supset M{:}\alpha$. Then:

(1) If σ is a type substitution then $\vdash \sigma(\Gamma \supset M{:}\,\alpha)$.

(2) For any type-assignment η, for any type substitution σ, and for any assignment ρ such that $\eta \uparrow_{\sigma\Gamma} \rho$, we have:

$$[\![\, \sigma(\Gamma \supset M{:}\,\alpha)\,]\!]\eta\rho \simeq [\![\, \Gamma \supset M{:}\alpha\,]\!]\eta'\rho$$

where: $\eta'(t) \triangleq [\![\, \sigma t\,]\!]\eta$.

4.3 Lemma *(Term Substitution)*

If $\vdash \Gamma, x{:}\alpha \supset M{:}\beta$, and $\vdash \Gamma \supset N{:}\alpha$ then

(1) $\vdash \Gamma \supset [N/x]M{:}\beta$

(2) For any type-assignment η, for any assignment ρ such that $\eta \uparrow_\Gamma \rho$,

$$[\![\Gamma \supset N{:}\alpha]\!]\eta\rho \Downarrow \;\Rightarrow\; [\![\Gamma \supset [N/x]M{:}\beta]\!]\eta\rho \simeq [\![\Gamma, x{:}\alpha \supset M{:}\beta]\!]\eta\rho'$$

where: $\rho' \triangleq \rho[\,[\![\Gamma \supset N{:}\alpha]\!]\eta\rho\ /x\,]$.

The following is proved by induction on the structure of C.

4.4 Lemma *(Canonical Forms are Defined)*

If $\vdash \emptyset \supset C{:}\alpha$, where C is a canonical form, then for any η and ρ, $[\![\emptyset \supset C{:}\alpha]\!]\eta\rho \Downarrow$.

The following is proved by induction on the deduction of the evaluation judgment.

4.5 Lemma *(Invariance under Evaluation)*

If $\vdash M \mapsto C$ then, for any η and ρ, $[\![\emptyset \supset M{:}\alpha]\!]\eta\rho = [\![\emptyset \supset C{:}\alpha]\!]\eta\rho$.

5. Adequacy

We want to prove that given a well typed closed term M, $M\Downarrow$ iff $M\downarrow$. It is easy to show that if $M\downarrow$ then $M\Downarrow$, as the interpretation is invariant under evaluation (4.5), and the interpretation of a canonical form is defined (4.4). In the other direction an iterated attempt of generalizing the induction hypothesis leads to the following

5.1 Definition *(adequacy relation)*

Let η be a type assignment. A relation $S \subseteq |[\![\alpha]\!]\eta| \times \Lambda^o_\alpha$ is an adequacy relation of type α, w.r.t. the type assignment η, if it satisfies the following conditions:

(C.1) $h\,S\,M \Rightarrow M\downarrow$.

(C.2) $\{h_n\}_{n<\omega}$ directed in D $\wedge\ \forall n.\ h_n S\,M \Rightarrow (\sqcup_{n<\omega} h_n)\,S\,M$.

(C.3) $(h\,S\,M \wedge \vdash M \mapsto C \wedge \vdash M' \mapsto C) \Rightarrow h\,S\,M'$.

(C.4) $(h\,S\,M \wedge h\,[\![\alpha]\!]\eta\,h') \Rightarrow h'\,S\,M$.

We denote with $AR(\alpha, \eta)$ the collection of adequacy relations of type α, w.r.t. the type assignment η. Observe that, for any type α, the empty set is an adequacy relation of type α.

If one thinks of h as an element in the equivalence class corresponding to the interpretation of the term M, then condition (C.1) corresponds to what we need to prove, i.e. $M\Downarrow \Rightarrow M\downarrow$. Condition (C.2) says that an adequacy relation is a kind of admissible predicate, in that it is closed under directed sets. Condition (C.3) says that an adequacy relation is invariant w.r.t terms that reduce to the same canonical form. The formulation of this condition seems to be new; it has the advantage of being simple and of not requiring a finer analysis of the evaluation relation as the closure of a "one-step" reduction relation. Condition (C.4) says that an adequacy relation is invariant w.r.t. the equivalence induced by the corresponding per. This condition comes from the choice of representing adequacy relations as relations over the field of a per rather than over the collection of its equivalence classes .

We wish to assign to each type α an adequacy relation, parametrically in a type assignment η. In order to do this correctly we need to introduce two further parameters:

(i) A *type substitution* σ.

(ii) An *adequacy relation assignment*, θ: $tv \to \cup_{\alpha\ type} AR(\alpha, \eta)$, (henceforth ar-assignment) that depends on η.

Let η be a type assignment. A substitution σ and an ar-assignment θ are *compatible*, w.r.t η, if $\theta(t) \in AR(\sigma t, \eta)$, for any t. We write this fact as: $\sigma \uparrow_\eta \theta$.

Proviso *(on bounded variables and substitution)*

In order to simplify the notation, in the following definition and proofs, whenever we bring a substitution under a bound variable we assume that the bound variable has been suitably redenominated so that it does not interact with the substitution.

5.2 Definition *(associating adequacy relations to types)*

Let α be any type. For any type assignment η, any substitution σ, and any ar-assignment θ, such that $\sigma \uparrow_\eta \theta$, we define a relation

$$R(\alpha, \sigma, \theta) \subseteq |[\![\sigma\alpha]\!]\eta| \times \Lambda^o_{\sigma\alpha}$$

by induction on the structure of α, as follows:

(1) $R(1, \sigma, \theta) \triangleq \{(h, M) \in D \times \Lambda^o_1 \mid M\downarrow\}$

(tv) $R(t, \sigma, \theta) \triangleq \theta(t)$

($\to$) $R(\alpha \to \beta, \sigma, \theta) \triangleq \{(h, M) \in |[\![\sigma(\alpha \to \beta)]\!]\eta| \times \Lambda^o_{\sigma(\alpha \to \beta)} \mid M\downarrow \wedge (d\, R(\alpha, \sigma, \theta)\, N \Rightarrow (hd\, R(\beta, \sigma, \theta)\, MN \vee hd\Uparrow))\}$

($\forall$) $R(\forall t.\alpha, \sigma, \theta) \triangleq \{(h, M) \in |[\![\sigma(\forall t.\alpha)]\!]\eta| \times \Lambda^o_{\sigma(\forall t.\alpha)} \mid M\downarrow \wedge \forall\beta.\, \forall S \in AR(\beta, \eta).\, h\, R(\alpha, [\beta/t]\sigma, \theta[S/t])\, M\beta\}$

5.3 Lemma *(coherence of the definition)*

Let α be any type. For any type assignment η, any substitution σ, and any ar-assignment θ, such that $\sigma \uparrow_\eta \theta$, we have: $R(\alpha, \sigma, \theta) \in AR(\sigma\alpha, \eta)$.

Proof

By induction on the structure of the type α we verify that $R(\alpha, \sigma, \theta)$ is well defined and belongs to $AR(\sigma\alpha, \eta)$ as it satisfies (C.1-4).

(1) $R(1, \sigma, \theta) \in AR(1, \eta)$, as one can easily verify that $\{(h, M) \in D \times \Lambda^o_1 \mid M\downarrow\}$ satisfies (C.1-4).

(tv) $R(t, \sigma, \theta) = \theta(t) \in AR(\sigma t, \eta)$, by the assumption: $\sigma \uparrow_\eta \theta$.

($\to$) By definition $R(\alpha \to \beta, \sigma, \theta) \subseteq |[\![\sigma(\alpha \to \beta)]\!]\eta| \times \Lambda^o_{\sigma(\alpha \to \beta)}$, and satisfies (C.1).

(C.2): Suppose $\{h_n\}_{n<\omega}$ directed, and $\forall n.\ h_n R(\alpha\rightarrow\beta, \sigma, \theta)\ M$. Then:
(i) $\sqcup_{n<\omega} h_n \in |[\![\sigma(\alpha\rightarrow\beta)]\!]\eta|$, because $[\![\sigma(\alpha\rightarrow\beta)]\!]\eta \in$ dcper.
(ii) For any $d \in D$, $(\sqcup_{n<\omega} h_n)d \simeq \sqcup_{n<\omega} h_n d$, and $\{h_n d \mid h_n d\Downarrow\}_{n<\omega}$ is directed, if not empty.
(iii) Suppose $d\ R(\alpha, \sigma, \theta)\ N$. There are two possibilities:
(a) $(\sqcup_{n<\omega} h_n)d \Uparrow$, and we are done.
(b) $(\sqcup_{n<\omega} h_n)d \Downarrow$, that implies $h_m d\Downarrow$, for some m, and therefore $h_n d\ R(\beta, \sigma, \theta)\ MN$, for all $n\geq m$. We can then apply the ind. hyp. on β to conclude $(\sqcup_{n<\omega} h_n d)\ R(\beta, \sigma, \theta)\ MN$, i.e. by (i, ii) $(\sqcup_{n<\omega} h_n)d\ R(\beta, \sigma, \theta)\ MN$.
(iv) By combining cases (iii.a-b) we have: $\sqcup_{n<\omega} h_n R(\alpha\rightarrow\beta, \sigma, \theta)\ M$.

(C.3): Suppose $(h\ R(\alpha\rightarrow\beta, \sigma, \theta)\ M \wedge \vdash M\mapsto C \wedge \vdash M'\mapsto C)$. Then $h\ R(\alpha\rightarrow\beta, \sigma, \theta)\ M'$ because:
(i) $M' \in \Lambda^o{}_{\sigma(\alpha\rightarrow\beta)}$ and $M'\downarrow$ (as the evaluation preserves the type).
(ii) Given N, $\vdash MN\mapsto C'$ iff $\vdash M'N\mapsto C'$.
(iii) Suppose $d\ R(\alpha, \sigma, \theta)\ N$. By (ii): $hd\Uparrow \vee (hd\Downarrow \wedge MN\downarrow \wedge M'N\downarrow)$. In the first case we are done, in the latter one shows: $hd\ R(\beta, \sigma, \theta)\ M'N$ by applying the inductive hypothesis on β, and (C.3) with (ii).

(C.4): Suppose $(h\ R(\alpha\rightarrow\beta, \sigma, \theta)\ M \wedge h\ [\![\sigma(\alpha\rightarrow\beta)]\!]\eta\ h')$.
Then, by definition of pexp: $d\ [\![\sigma\alpha]\!]\eta\ d' \Rightarrow (hd\ [\![\sigma\beta]\!]\eta\ h'd' \vee (hd\Uparrow \wedge h'd\Uparrow))$.
To show $h'\ R(\alpha\rightarrow\beta, \sigma, \theta)\ M$ we have to verify:

$d\ R(\alpha, \sigma, \theta)\ N \Rightarrow (h'd\ R(\beta, \sigma, \theta)\ MN \vee h'd\Uparrow)$.

But: $h'd\Uparrow \Leftrightarrow hd\Uparrow$, and $h'd\Downarrow \Rightarrow hd\Downarrow \Rightarrow (hd\ R(\beta, \sigma, \theta)\ MN)$
$\Rightarrow (h'd\ R(\beta, \sigma, \theta)\ MN)$, the last implication by ind. hyp. on β and (C.4).

($\forall$) By definition $R(\forall t.\alpha, \sigma, \theta) \subseteq |[\![\sigma(\forall t.\alpha)]\!]\eta| \times \Lambda^o{}_{\sigma(\forall t.\alpha)}$ and satisfies (C.1).
Observe that for any type β: $(\sigma\uparrow_\eta\theta \wedge S\in AR(\beta, \eta)) \Rightarrow [\beta/t]\sigma\uparrow_\eta\theta[S/t]$.

(C.2) Suppose $\{h_n\}_{n<\omega}$ directed, and $\forall n.\ h_n R(\forall t.\alpha, \sigma, \theta)\ M$. Then,

$\forall\beta.\ \forall S\in AR(\beta, \eta).\ h_n\ R(\alpha, [\beta/t]\sigma, \theta[S/t])\ M\beta$

Hence, by induction hypothesis on α, and (C.2):

$\forall\beta.\ \forall S\in AR(\beta, \eta).\ \sqcup_{n<\omega} h_n\ R(\alpha, [\beta/t]\sigma, \theta[S/t])\ M\beta$

and this implies by definition of $R(\forall t.\alpha, \sigma, \theta)$:

$\sqcup_{n<\omega} h_n\ R(\forall t.\alpha, \sigma, \theta)\ M$

(C.3): Suppose $(h\ R(\forall t.\alpha, \sigma, \theta)\ M \wedge \vdash M\mapsto C \wedge \vdash M'\mapsto C)$.
Then $h\ R(\forall t.\alpha, \sigma, \theta)\ M'$ because:
(i) $M'\in \Lambda^o{}_{\sigma(\forall t.\alpha)}$ and $M'\downarrow$.
(ii) For any β, $\vdash M\beta\mapsto C'$ iff $\vdash M'\beta\mapsto C'$.
(iii) For any β, for any $S\in AR(\beta, \eta)$, we can apply (C.3) on α (using (ii) and ind. hyp.), with substitution $[\beta/t]\sigma$, and ar-assignment $\theta[S/t]$ to conclude:
$h\ R(\alpha, [\beta/t]\sigma, \theta[S/t])\ M'\beta$. Hence: $h\ R(\forall t.\alpha, \sigma, \theta)\ M'$.

(C.4): Suppose $(h\ R(\forall t.\alpha, \sigma, \theta)\ M \wedge h\ [\![\sigma(\forall t.\alpha)]\!]\eta\ h')$.
Then, by definition: $\forall B$ dcper. $h\ [\![\sigma\alpha]\!]\eta[B/t]\ h'$.
To show $h'\ R(\forall t.\alpha, \sigma, \theta)\ M$ we have to verify:

$\forall\beta.\ \forall S\in AR(\beta,\eta).\ h'\ R(\alpha, [\beta/t]\sigma, \theta[S/t])\ M\beta$.
But for $B = [\![\beta]\!]\eta$, we have $[\![\sigma\alpha]\!]\eta[B/t] = [\![[\beta/t]\sigma\alpha]\!]\eta$, by the type susbstitution lemma. Hence $h\ [\![[\beta/t]\sigma\alpha]\!]\eta\ h'$, and we can apply ind. hyp. on α, and (C.4). □

5.4 Theorem *(semantic and syntactic convergence coincide)*

Suppose $\vdash (x_1: \alpha_1),\ldots,(x_n: \alpha_n) \supset M: \alpha$. Then for any type assignment η, any substitution σ, and any ar-assignment θ, such that $\sigma\uparrow_\eta\theta$, we have:

if $d_i\ R(\alpha_i, \sigma, \theta)\ C_i$, for i=1,..,n, and $[\![\sigma(\Gamma \supset M: \alpha)]\!]\eta[d/x] = [h]_A$
then $h\ R(\alpha, \sigma, \theta)\ [C/x]\sigma M$.

where: $\Gamma \equiv (x_1: \alpha_1),\ldots,(x_n: \alpha_n)$; $[d/x] \equiv [\,[d_1]_{A_1}/x_1,\ldots,[d_n]_{A_n}/x_n]$; $A_i \equiv [\![\sigma\alpha_i]\!]\eta$ for i=1,..,n; $A \equiv [\![\sigma\alpha]\!]\eta$; $[C/x] \equiv [C_1/x_1,\ldots, C_n/x_n]$.

Proof

In order to have a quick understanding of the statement observe, as an instance: if $\vdash \emptyset \supset M: \alpha$ where M and α have no free type variables, and $[\![\emptyset \supset M: \alpha]\!]\eta\rho = [h]_A$ then $hR(\alpha, id, \theta)M$, for arbitrary η, ρ, θ. Hence, by (C.1), we have: $M\Downarrow\ \Rightarrow\ M\downarrow$.

We now proceed with the proof, by induction on the length of the typing.

(*) Then $[\![\sigma\Gamma \supset *: 1]\!]\eta[d/x] = [d]_1$, and $d\ R(1, \sigma, \theta)\ *$, by def. of $R(1, \sigma, \theta)$.

(asmp) Then $[\![\sigma\Gamma \supset x_i: \sigma\alpha_i]\!]\eta[d/x] = [d_i]_{A_i}$, and $d_i\ R(\alpha_i, \sigma, \theta)\ C_i$, by assumption.

($\to$I) Suppose $h\in[\![\sigma(\Gamma \supset \lambda x{:}\alpha.M{:}\alpha\to\beta)]\!]\eta[d/x]$. Then by the interpretation definition: $\forall d\in |A|.\ (f(d)\Downarrow \Rightarrow hd\in f(d)) \wedge (f(d)\Uparrow \Rightarrow hd\Uparrow)\}$ where: $A = [\![\sigma\alpha]\!]\eta$,
$f(d) = [\![\sigma(\Gamma,x{:}\alpha \supset M{:}\beta)]\!]\eta[d/x][[d]_A/x]$.
We have to show: $h\ R(\alpha\to\beta, \sigma, \theta)\ [C/x]\sigma(\lambda x{:}\alpha.M)$, that is:
(i) $[C/x]\sigma(\lambda x{:}\alpha.M)\downarrow$, that holds because an abstraction always converges.
(ii) $d\ R(\alpha, \sigma, \theta)\ N\ \Rightarrow\ (hd\ R(\beta, \sigma, \theta)\ [C/x]\sigma(\lambda x{:}\alpha.M)\ N) \vee (hd\Uparrow)$.
Observe: $[C/x]\sigma(\lambda x{:}\alpha.M) \equiv \lambda x{:}\sigma\alpha.[C/x]\sigma M$.
Also: $\vdash (\lambda x{:}\sigma\alpha.[C/x]\sigma M)N\mapsto C$ iff $\vdash [C'/x][C/x]\sigma M\mapsto C$ and $\vdash N\mapsto C'$.
Suppose $hd\Downarrow$, we can apply the inductive hypothesis on the provable judgment $\Gamma,x{:}\alpha \supset M{:}\beta$ to get $hd\ R(\beta, \sigma, \theta)\ [C'/x][C/x]\sigma M$. Hence by (C.1) $[C'/x][C/x]\sigma M\downarrow$, *and by* (C.3) we get (ii).

($\to$E) Suppose $d\in[\![\sigma(\Gamma\supset MN{:}\beta)]\!]\eta[d/x]$ then we have to show $d\ R(\beta, \sigma, \theta)\ [C/x]\sigma(MN)$. By the definition of the interpretation we have $h\in[\![\sigma(\Gamma\supset M{:}\alpha\to\beta)]\!]\eta[d/x]$, and $d'\in[\![\sigma(\Gamma\supset N{:}\alpha)]\!]\eta[d/x]$, such that $hd'[\![\sigma\beta]\!]\eta\ d$.
By induction hypothesis we have: $h\ R(\alpha\to\beta, \sigma, \theta)\ [C/x]\sigma(M)$, and
$d'\ R(\alpha, \sigma, \theta)[C/x]\sigma(N)$. *By definition of* $R(\alpha\to\beta, \sigma, \theta)$ we can conclude:
$hd'\ R(\beta, \sigma, \theta)\ [C/x]\sigma(MN)$, and by (C.4) $d\ R(\beta, \sigma, \theta)\ [C/x]\sigma(MN)$.

($\forall$I) Suppose $h\in[\![\sigma(\Gamma\supset\lambda t.M{:}\forall t.\alpha)]\!]\eta[d/x]$ then we have to show:
(i) $[C/x]\sigma(\lambda t.M)\downarrow$
(ii) $\forall\beta.\ \forall S\in AR(\beta,\eta).\ h\ R(\alpha, [\beta/t]\sigma, \theta[S/t])\ [C/x]\sigma(\lambda t.M)\beta$.
Observe: $[C/x]\sigma(\lambda t.M) \equiv \lambda t.[C/x]\sigma M$. By induction hypothesis applied to $\Gamma\supset M{:}\ \alpha$, we have, $\forall\beta.\ \forall S\in AR(\beta,\eta) : h\ R(\alpha, [\beta/t]\sigma, \theta[S/t])\ [C/x][\beta/t]\sigma M$.
Observe: $\vdash [C/x]\sigma(\lambda t.M)\beta \mapsto C'$ iff $\vdash [C/x][\beta/t]\sigma M \mapsto C'$, *and apply* (C.3).

($\forall$E) Suppose $h \in [\![\sigma(\Gamma \supset M\beta:[\beta/t]\alpha)]\!]\eta[d/x]$, then we have to show:
$h\ R([\beta/t]\alpha, \sigma, \theta)\ [C/x]\sigma(M\beta)$.
By induction hypothesis applied to $\Gamma \supset M$: $\forall t.\alpha$, and the *definition of* $R(\forall t.\alpha, \sigma, \theta)$ we have: $h\ R(\alpha, [\sigma\beta/t]\sigma, \theta[S/t])\ [C/x]\sigma M\sigma\beta$, and observe $\sigma[\beta/t]\alpha = [\sigma\beta/t]\sigma\alpha$.

(Y) Suppose $[\ \sqcup_{n<\omega} k(n)\ ]_A = [\![\sigma(\Gamma \supset YM:\alpha)]\!]\eta[d/x]$, where: $k \in [\![\sigma(\Gamma \supset M:(1 \rightharpoonup \alpha) \rightharpoonup \alpha)]\!]\eta[d/x]$ for some k. We show that for any n big enough $k(n)\ R(\alpha, \sigma, \theta)\ [C/x]\sigma YM$. We conclude *by (C.2)*: $\sqcup_{n<\omega} k(n)\ R(\alpha, \sigma, \theta)\ [C/x]\sigma YM$. By induction hypothesis on $\Gamma \supset M$: $(1 \rightharpoonup \alpha) \rightharpoonup \alpha$ we know: $k\ R((1 \rightharpoonup \alpha) \rightharpoonup \alpha, \sigma, \theta)\ [C/x]\sigma M$. Observe:

(i) $i(\lambda d \in D.\Uparrow)\ R((1 \rightharpoonup \alpha), \sigma, \theta)\ \lambda x{:}1.[C/x]\sigma YM$.

(ii) $i(\lambda d \in D.k(n))\ R((1 \rightharpoonup \alpha), \sigma, \theta)\ \lambda x{:}1.[C/x]\sigma YM \Rightarrow$
$k\ i(\lambda d \in D.k(n))\ R(\alpha, \sigma, \theta)\ [C/x]\sigma M\ (\lambda x{:}1.[C/x]\sigma YM) \vee k(n+1)\Uparrow$.

Use (C.3) to conclude $k(n)\ R(\alpha, \sigma, \theta)\ [C/x]\sigma YM$, for n big enough. □

Note *(on the generalization of the inductive hypothesis)*

We have emphasized in italic the points in the proof above where one discovers the need to generalize the inductive hypothesis, and to adapt the definition of R.

(1) The need of working with a judgment, $\Gamma \supset M:\alpha$, where Γ may be a non empty context, already appears at the ($\rightharpoonup$I) case. Here also appears the need for some condition of closure under expansions, this problem also arises in the cases ($\forall$I), and (Y). How to formalize this condition is a tricky point, the simplest solution we could find is formalized as condition (C.3).

(2) In the case ($\rightharpoonup$E) it becomes evident that an adequacy relation associated to an $\rightharpoonup$ type should satisfy certain functional properties. This leads to clause ($\rightharpoonup$) of definition 5.2. A similar problem arises when dealing with second order types. Here however we hit the problem of resolving a potential circularity in definition 5.2. Hence the need to introduce an abstract notion of adequacy relation and the notion of ar-assignment. Substitution are introduced in 5.2 to ensure that the definition is "well-typed". We emphasize that similar problems arise in the context of proofs of (strong) normalization of Girard system F (see, e.g., Girard&al[89], chpt. 14).

(3) Finally the way the fixed point is computed in the interpretation naturally suggests the introduction of (C.2) in order to deal with clause (Y).

6. Conclusion

(1) *Relating model-theoretic and operational pre-orders*

We write $\Gamma \supset M \leq_{obs} N$: α, if $\vdash \Gamma \supset M$: α, $\vdash \Gamma \supset N$: α, and for any "context" C[ ] such that $\vdash \varnothing \supset C[M]$: β, and $\vdash \varnothing \supset C[N]$: β, we have: $C[M]\downarrow \Rightarrow C[N]\downarrow$.

This defines an *operational* preorder on terms. In the standard domain theoretic case it is easy to prove, as a corollary of the correspondence between syntactic and semantic convergence, that the pre-order induced by the model (definition left to the reader) is contained in the operational preorder defined above.

When considering per-models there is a natural *intrinsic* way (Rosolini[86]) to order the equivalence classes:

Let A be a per over D. We define the intrinsic preorder $\leq_A$ over [A] as follows: $x \leq_A y$ if $\forall h: A \rightharpoonup 1.\ (h(x){\Downarrow} \Rightarrow h(y){\Downarrow})$

How does the intrinsic pre-order relate to the operational pre-order ? We expect that the former is included in the latter, but to have a proof as simple as in the domain-theoretic case some additional model-theoretic information seems needed. In particular one needs to show that contexts induce monotone operators w.r.t. the intrinsic pre-order, and this seems to depend on the fact that products, used in the intepretation of second order types, have a pointwise intrinsic preordering. We do not have such a property available for the model described here. There are other per models that enjoy this property, e.g. the model of complete extensional pers over Kleene partial combinatory algebra (Freyd&al.[90]). We expect that the "architecture" of the proof of our main result (5.4) can be applied to such per-models. Hence we suspect that for such models a standard proof of the theorem relating intrinsic and operational pre-ordering will go through.

(2) *Call-by-name, Subtyping, and Recursive Types*

It seems worthwhile to recall that a call-by-name version of the calculus can be easily coded in the calculus presented here with the standard idea that the call-by-name functional space, say $\alpha \rightharpoonup_n \beta$, is coded as $(1 \rightharpoonup \alpha) \rightharpoonup \beta$. We also recall that per-models have been used as a semantic foundation for typed functional languages with a notion of *subtyping*(Cardelli&Longo[90]). Our result suggests that this is an *adequate* approach.

We expect that our main result extends to recursive types once we take as semantic structure the collection of complete "uniform" pers over a D-infinity model, as presented in Amadio[89]. The basic idea is to "stratify" the definition of the adequacy relation associated to a type. Formally one introduces an intermediary family of adequacy relation $R(n, \alpha, \sigma, \theta)$, where $n \in \omega$. The adequacy relation $R(\alpha, \sigma, \theta)$ is obtained by a process of completion of the sequence $\{R(n, \alpha, \sigma, \theta) \mid n \in \omega\}$. Roughly $R(n, \alpha, \sigma, \theta)$ represents $R(\alpha, \sigma, \theta)$ cut at the n-th level of the construction of the underlying D-infinity structure. We refrain from going into this point since the development of the model requires a certain number of rather ad hoc conditions that would only obscure the main ideas we have discussed.

(3) *Independence from the Adequacy of the Realizability Structure*

It is well know that to every, say closed, term M of type α one can associate its "erasure", i.e. a type free term er(M) such that: $[\![M]\!]^{per} \cong [\, [\![er(M)]\!]^D]_{[A]}$ (a).

In Amadio[90] we suggested that a "cheap" adequacy theorem could be obtained by the following schema of implications:

$$[\![M]\!]^{per}{\Downarrow} \Rightarrow_{(1)} [\![er(M)]\!]^D{\Downarrow} \Rightarrow_{(2)} er(M){\downarrow} \Rightarrow_{(3)} M{\downarrow}$$

where (1) follows by a result of type (a), (2) follows by the adequacy of the realizability structure, and (3) follows by a comparison of the evaluations. The weak point of this chain of implications is (2). As a matter of fact we have shown that the adequacy of the per model is *independent* from the adequacy of the realizability structure w.r.t. the related type-free language. Following Baeten&Boerboom[79] one can build a realizability structure D that is *not* adequate in that $[\Omega]^D \Downarrow$, where $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$. Of course one can still hope to show (2) for terms coming from the erasure of a typed term. But then we have basically to reprove the same result presented here following a "type-assignment" style.

References

Amadio R. [1989] "Recursion over realizability structures", Info.&Comp., 91, 1, (55-85), also appeared as TR 1/89 Dipartimento di Informatica, Università di Pisa.

Amadio R. [1990] "Domains in a Realizability Framework", in Proc. CAAP91, Brighton, SLNCS 493, Abramsky S., Maibaum T. (eds.), (241-263), full version appeared as Liens TR 19-90, Paris.

Amadio R., Cardelli L. [1990] "Subtyping Recursive Types", in Proc. ACM-POPL91, Orlando, full version appeared as DEC-SRC TR #62, Palo Alto.

Baeten J, Boerboom B. [1979] "Ω can be anything it shouldn't be", Indag. Math., 41, (111-120).

Cardelli L. [1989] "The Quest language and system", pre-print DEC-SRC, Palo Alto.

Cardelli L., Longo G.[1990] "A semantic basis for Quest", in Proc. ACM-Lisp and Functional Programming 90, Nice.

Freyd P., Mulry P., Rosolini G., Scott D. [1990] "Extensional Pers", in Proc. 5th IEEE-LICS, Philadelphia.

Girard J.Y., Lafont Y., Taylor P. [1989] "Proofs and Types", Cambridge University Press.

Hyland M. [1991] "First steps in synthetic domain theory", in Proc. Category Theory 90, Carboni&al. (eds.), Springer-Verlag, (to appear).

Martin-Löf [1983] "The domain interpretation of type theory", in Proc. Workshop on the Semantics of Programming Languages, Dybier&al (eds.), Chalmers University, Göteborg.

Moggi E. [1988] "Partial morphisms in categories of effective objects", Info.&Comp., 76, (250-277).

Plotkin G. [1985] "Denotational semantics with partial functions", lecture notes, CSLI, Stanford 1985.

Rosolini G. [1986] "Continuity and effectiveness in Topoi", PhD Thesis, Oxford University.

Nancy, Wed, Dec 11, 1991.

Appendix A

In this appendix we present the proofs of the results stated in the first four sections of the paper.

Properties Semantic Structure *(Proposition 3.3)*

(1) **dcpper** is a pccc.

(2) dcpers are closed under arbitrary intersections.

(3) **dcpper** is reflective in **pper**.

(4) **dcpper** has fixpoints over objects of the shape pexp(**1**,A). Such fixpoints are the least ones (up to equivalence) w.r.t. the intrinsic pre-order.

Proof

We sketch the main ideas of the techniques described in Amadio[89], Amadio[90].

(1) It is enough to verify that (i) **1** is a dcper and that (ii) $A\times B$, $A \rightharpoonup B$ are dcpers if A, and B are dcpers.

(2) Immediate.

(3) This means that the inclusion functor from **dcper** to **pper** has a left adjoint, say L: **pper**→**dcpper**. Define $L(A) = \bigcap \{B \in \text{dcper} \mid A \subseteq B\}$. This is well defined because of (2). Next show: $\forall A \in$ per. $\forall B \in$ dcper. pexp(A, B) = pexp(L(A), B). Hint: this can be proved by induction once it has been observed that L(A) can be characterized as the closure of a certain operator G over A.

Namely define: $G(A') = TC(Sup(A'))$, for A' per

where: $Sup(A') = \{\sqcup X \mid X \subseteq A', X \text{ directed}\}$, and TC(A') is the transitive closure of A'.

Then inductively: $G_0(A) = A$, $G_{\alpha+1}(A) = G(G_\alpha(A))$,

$G_\lambda(A) = \bigcup_{\alpha<\lambda} G_\alpha(A)$ (for λ limit ordinal).

Hence $L(A) = G_\beta(A)$ for some β, and one shows pexp(A, B) = pexp(G_α(A), B) by induction.

(4) Define $Fix = i(\lambda k \in D.\ \sqcup_{n<\omega} k(n))$, where $k(0) \cong \Uparrow$, $k(n+1) \cong k\ i(\lambda d \in D.\ k(n))$.

One shows: for any **dcpper** A, Fix $\in$ | pexp(B, A) |,

where B = pexp(pexp(**1**,A),A).

Observe Fix k $\cong \sqcup_{n<\omega} k(n)$. Show by induction on n that:

$$k\ B\ k' \Rightarrow k(n)\ A\ k'(n) \ \vee\ (k(n)\Uparrow \wedge k'(n)\Uparrow).$$

Since A is a dcper conclude: $\sqcup_{n<\omega}k(n)\ A\ \sqcup_{n<\omega}k'(n) \ \vee\ (\sqcup_{n<\omega}k(n)\Uparrow \wedge \sqcup_{n<\omega}k'(n)\Uparrow)$. □

Typing Soundness *(Proposition 4.1)*

If $\vdash \Gamma \supset M:\alpha$ then, for any η and ρ such that $\eta \uparrow_\Gamma \rho$, we have:

$$[\![\Gamma \supset M:\alpha]\!]\eta\rho \Downarrow \ \Rightarrow\ [\![\Gamma \supset M:\alpha]\!]\eta\rho \in [\ [\![\alpha]\!]\eta\ ].$$

A direct proof of this result by induction on the length of the typing judgment does not seem to work as we hit the problem of showing in the case ($\rightharpoonup$I), and ($\forall$I) that certain collections of realizers are non-empty. Hence, it appears that a natural argument to prove the existence of such realizers is to consider the interpretation in the realizability structure of an underlying type-free lambda term.

The function *er* (erasure) takes a typed term and returns a type-free term in the language generated by the following BNF (as usual v stands for the variables):

$$P ::= * \mid v \mid (\lambda v.P) \mid (PP) \mid (Y_V\, P)$$

It is defined by induction on the structure as follows:

$$er(*) = *;\ \ er(x) = x;\ \ er(\lambda x{:}\alpha.M) = (\lambda x.er(M));\ \ er(MN) = (er(M)er(N));$$
$$er(\lambda t.M) = er(M);\ \ er(M\beta) = er(M);\ \ er(YM) = (Y_V er(M))$$

Given a realizability structure (D, i, j) as in 3.1 we can define a standard interpretation of the lambda calculus introduced above. We denote with τ a partial function from variables to D. Define:

(*) $[\![*]\!]\tau = d$ for some fixed $d\in D$ (D supposed non-empty !)

(asmp) $[\![x]\!]\tau \simeq \tau(x)$

(abs) $[\![\lambda x.P]\!]\tau = i(\lambda d\in D.[\![P]\!]\tau[d/x])$.

(apl) $[\![PQ]\!]\tau \simeq j([\![P]\!]\tau)([\![Q]\!]\tau)$

(Y_V) $[\![Y_V P]\!]\tau \simeq \bigsqcup_{n<\omega} h(n)$

where: $h \simeq [\![P]\!]\tau,\ \ h(0) \simeq \Uparrow,\ \ h(n+1) \simeq h\, i(\lambda d\in D.h(n))$.

Given a context Γ, a type assignment η, an assignment ρ s.t. $\eta\uparrow_\Gamma\rho$, and an assignment τ, we say that ρ is compatible with τ w.r.t. Γ, and write $\rho\uparrow_\Gamma\tau$, if for any variable x in Γ: $(\rho(x)\Downarrow\ \Leftrightarrow\ \tau(x)\Downarrow)\ \wedge\ (\rho(x)\Downarrow\ \Rightarrow\ \tau(x)\in\rho(x))$.

We can now state a proposition stronger than 4.1.

Proposition *(erasure)*

Suppose $\vdash\Gamma\supset M{:}\alpha$. Then for any η, for any ρ s.t. $\eta\uparrow_\Gamma\rho$, for any τ s.t. $\rho\uparrow_\Gamma\tau$:

(1) $[\![\Gamma\supset M{:}\alpha]\!]\eta\rho\Downarrow\ \Leftrightarrow\ [\![er(M)]\!]\tau\Downarrow$.

If $[\![\Gamma\supset M{:}\alpha]\!]\eta\rho\Downarrow$ then:

(2) $[\![er(M)]\!]\tau \in |A|$, where $A = [\![\alpha]\!]\eta$.

(3) $[\![\Gamma\supset M{:}\alpha]\!]\eta\rho = [\,[\![er(M)]\!]\tau\,]_A$.

Proof

By induction on the length of the typing. We refer to the notation in the interpretation definition.

(*) (1) $[\![\Gamma\supset *{:}\,1]\!]\eta\rho = [d]_1$, $[\![er(M)]\!]\tau = d$. (2) $d\in D$, $D = |[\![1]\!]\eta|$. (3) $d\in[d]_1$.

(asmp) (1-3) By definition of $\rho\uparrow_\Gamma\tau$.

($\rightharpoonup$I) (1) Both interpretations are always defined.

(2) Use ind. hyp. on $\Gamma, x{:}\alpha\supset M{:}\beta$ to show:

$$d\,A\,e\ \Rightarrow\ ([\![er(M)]\!]\tau[d/x]\Uparrow\ \wedge\ [\![er(M)]\!]\tau[e/x]\Uparrow) \vee ([\![er(M)]\!]\tau[d/x]\ B\ [\![er(M)]\!]\tau[e/x]).$$

(3) Let $x = \{h\in D \mid \forall d\in|A|.\ (fd\Downarrow\ \Rightarrow hd\in f(d)) \wedge (fd\Uparrow\ \Rightarrow hd\Uparrow)\}$. We show:

(i) $h, h'\in x\ \Rightarrow\ h\ [\![\alpha\rightharpoonup\beta]\!]\eta\ h'$, and (ii) $h\ [\![\alpha\rightharpoonup\beta]\!]\eta\ h'\ \wedge\ h\in x\ \Rightarrow\ h'\in x$. Apply ind. hyp. on f(d). (i) Observe: $d\ [\![\alpha]\!]\eta\ d'\ \Rightarrow\ (f(d)\Downarrow\ \Rightarrow hd, hd'\in f(d)) \wedge (f(d)\Uparrow\ \Rightarrow hd\Uparrow \wedge hd'\Uparrow)$.

(ii) Observe: $\forall d\in|A|.\ (f(d)\Downarrow\ \Rightarrow\ hd, h'd\in f(d)) \wedge (f(d)\Uparrow\ \Rightarrow hd\Uparrow \wedge h'd\Uparrow)$.

Finally prove: $[\![\lambda x.er(M)]\!]\tau \in [\![\Gamma\supset\lambda x{:}\alpha.\ M{:}\beta]\!]\eta\rho$.

($\rightharpoonup$E) Left to the reader.

($\forall$I) (1) $[\![\Gamma\supset\lambda t.M:\forall t.\alpha]\!]\eta\rho\Downarrow \Leftrightarrow \forall A.(f(A)\Downarrow) \Leftrightarrow [\![er(M)]\!]\hbar\Downarrow \Leftrightarrow [\![er(\lambda t.M)]\!]\hbar\Downarrow$.
(2) If $\forall A.\ [\![er(M)]\!]\hbar \in |[\![\alpha]\!]\eta[A/t]|$ then $[\![er(M)]\!]\hbar \in |\bigcap_{A\ dcper}[\![\alpha]\!]\eta[A/t]|$.
(3) Let $x = \{h\in D \mid \forall A$ dcper. $h\in f(A)\}$. We show:
(i) $h, h'\in x \Rightarrow h \bigcap_{A\ dcper}[\![\alpha]\!]\eta[A/t]\ h'$. (ii) $h \bigcap_{A\ dcper}[\![\alpha]\!]\eta[A/t]\ h', h\in x \Rightarrow h'\in x$.
(i) Since by ind. hyp., for any A dcper, $h, h'\in f(A)\in [\ [\![\alpha]\!]\eta[A/t]\ ]$. (ii) Since, for any A dcper, $h\ [\![\alpha]\!]\eta[A/t]\ h', h\in f(A) \Rightarrow h'\in f(A)$.

($\forall$E) Left to the reader.

(Y) Left to the reader, this requires an inductive argument of the type already seen for the (Y) clasuse. □

Note

Of course we could define a combinator Y_V directly in the type-free calculus as follows:

$$Y_V \equiv \lambda f.\ f\ (\lambda z.(\omega_V(f)\ \omega_V(f))),\ \ \omega_V(f) \equiv \lambda x.\ f(\lambda z.\ xx).$$

In order to have an intuition of Y_V behavior let us β-reduce $(Y_V\ f)$:

$$(Y_V\ f) \rightarrow_\beta f\ (\lambda z.(\omega_V(f)\ \omega_V(f))), \text{ and}$$
$$(\omega_V(f)\ \omega_V(f)) \equiv (\lambda x.\ f(\lambda z.\ xx)))\ \omega_V(f) \rightarrow_\beta f(\lambda z.\ (\omega_V(f)\ \omega_V(f)))$$

One may think of $Y_V\ f$ as: $f\ (\lambda z.f(\lambda z.\ f(\lambda z.\ f(\lambda z.\ ....))))$. The problem is that there is no reason why the interpretation of Y_V should behave like the interpretation given in the clause (Y_V) above, and we need such an intepretation to prove the "erasure" proposition! For instance taking the realizability structure described in Appendix B we have: $[\![\ Y_V\ (\lambda f.\ f *)\ ]\!]\hbar \Downarrow$, whereas we wish: $[\![\ Y\ (\lambda f{:}\ 1\rightharpoonup\alpha.\ f *)\ ]\!]\eta\rho \Uparrow$. □

Type Substitution *(Lemma 4.2)*

Suppose $\vdash\Gamma \supset M{:}\alpha$. Then:

(1) If σ is a type substitution then $\vdash \sigma(\Gamma \supset M{:}\ \alpha)$.

(2) For any type-assignment η, for any type substitution σ, and for any assignment ρ such that $\eta\uparrow_{\sigma\Gamma}\rho$, we have:

$$[\![\ \sigma(\Gamma \supset M{:}\ \alpha)\ ]\!]\eta\rho \cong [\![\ \Gamma \supset M{:}\alpha\ ]\!]\eta'\rho$$

where: $\eta'(t) \triangleq [\![\ \sigma t\ ]\!]\eta$.

Proof

Both assertions are proved by induction on the length of the typing.

(1) (*) If $\vdash\Gamma \supset *{:}\ 1$ then $\vdash\sigma\Gamma \supset *{:}\ 1$, since $\sigma\Gamma$ is a context.

(asmp) If $x{:}\ \alpha\in\Gamma$ then $x{:}\ \sigma\alpha\in\sigma\Gamma$.

($\rightharpoonup$I) If $\vdash\Gamma,x{:}\alpha \supset M{:}\beta$ then $\vdash\sigma\Gamma,x{:}\sigma\alpha \supset \sigma M{:}\sigma\beta$, by induction hypothesis. Hence: $\vdash\sigma\Gamma\supset \lambda x{:}\sigma\alpha.\ \sigma M{:}\ \sigma\alpha\rightharpoonup\sigma\beta \equiv \sigma(\Gamma\supset \lambda x{:}\alpha.\ M{:}\ \alpha\rightharpoonup\beta)$.

($\rightharpoonup$E) If $\vdash\Gamma \supset M{:}\ (\alpha\rightharpoonup\beta)$ and $\vdash\Gamma\supset N{:}\alpha$ then $\vdash\sigma\Gamma \supset \sigma M{:}\ (\sigma\alpha\rightharpoonup\sigma\beta)$ and $\vdash\sigma\Gamma\supset\sigma N{:}\sigma\alpha$. Hence: $\vdash\sigma\Gamma\supset(\sigma M\sigma N){:}\ \sigma\beta \equiv \sigma(\Gamma\supset(MN){:}\ \beta)$.

($\forall$I) Redenominate t so that it does not interfere with the substitution. If $\vdash\Gamma \supset M: \alpha$ and $t \notin ftv(\Gamma)$ then $\vdash \sigma(\Gamma \supset M: \alpha)$, by induction hypothesis. Hence $\vdash \sigma\Gamma \supset \lambda t.\sigma M: \forall t.\sigma\alpha$.

($\forall$E) If $\vdash \Gamma \supset M: (\forall t.\alpha)$ then $\vdash \sigma\Gamma \supset \sigma M: \forall t.\sigma\alpha$ for suitable t, by induction hypothesis. Hence: $\vdash \sigma\Gamma \supset \sigma M\sigma\beta: [\sigma\beta/t]\sigma\alpha \equiv \sigma(\Gamma \supset (M\beta): [\beta/t]\alpha)$.

(Y) If $\vdash \Gamma \supset M: (1 \rightharpoonup \alpha) \rightharpoonup \alpha$ then $\vdash \sigma\Gamma \supset \sigma M: (1 \rightharpoonup \sigma\alpha) \rightharpoonup \sigma\alpha$, by induction hypothesis. Hence: $\vdash \sigma\Gamma \supset Y_{\sigma\alpha}\sigma M: \sigma\alpha \equiv \sigma(\Gamma \supset Y_\alpha M: \alpha)$.

(2) Observe $\eta \uparrow_{\sigma\Gamma} \rho \Rightarrow \eta' \uparrow_\Gamma \rho$, as for any $(x{:}\alpha)$ in Γ: $[\![\sigma\alpha]\!]\eta = [\![\alpha]\!]\eta'$.

(*) $[\![\sigma(\Gamma \supset *: 1)]\!]\eta\rho = [d]_1 = [\![\Gamma \supset *: 1]\!]\eta'\rho$.

(asmp) $[\![\sigma(\Gamma \supset x: \alpha)]\!]\eta\rho \simeq \rho(x) \simeq [\![\Gamma \supset x: \alpha]\!]\eta'\rho$.

($\rightharpoonup$I) Observe $\sigma(\Gamma, x{:}\alpha \supset M{:}\beta) = \sigma\Gamma, x{:}\sigma\alpha \supset \sigma M{:}\sigma\beta$. For any $d \in |[\![\sigma\alpha]\!]\eta| = |[\![\alpha]\!]\eta'|$, we have by ind. hyp.: $[\![\sigma\Gamma, x{:}\sigma\alpha \supset \sigma M{:}\sigma\beta]\!]\eta\rho[[d]_A/x] \simeq [\![\Gamma, x{:}\alpha \supset M{:}\beta]\!]\eta'\rho[[d]_A/x]$. Hence by the definition of the interpretation:

$$[\![\sigma(\Gamma \supset \lambda x{:}\alpha.M: \alpha \rightharpoonup \beta)]\!]\eta\rho = [\![\Gamma \supset \lambda x{:}\alpha.M: \alpha \rightharpoonup \beta]\!]\eta'\rho.$$

($\rightharpoonup$E) Immediate application of the inductive hypothesis.

($\forall$I) As usual we assume that the bound variable t has been suitably redenominated. Then for $t \notin ftv(\Gamma)$ we have: $\eta[A/t] \uparrow_{\sigma\Gamma} \rho$, for any A. Hence by ind. hyp.: $[\![\sigma\Gamma \supset \sigma M{:}\sigma\alpha]\!]\eta[A/t]\rho \simeq [\![\Gamma \supset M{:}\alpha]\!]\eta'[A/t]\rho$. By the definition of the interpretation: $[\![\sigma(\Gamma \supset \lambda t.M{:}\forall t.\alpha)]\!]\eta\rho \simeq [\![\Gamma \supset \lambda t.M{:}\forall t.\alpha]\!]\eta'\rho$.

($\forall$E) Apply ind. hyp. observing: $[\![\sigma([\beta/t]\alpha)]\!]\eta = [\![[\beta/t]\alpha]\!]\eta'$.

(Y) Again an easy application of the ind. hyp. . $\square$

Term Substitution *(Lemma 4.3)*

If $\vdash\Gamma, x{:}\alpha \supset M{:}\beta$, and $\vdash\Gamma \supset N{:}\alpha$ then

(1) $\vdash\Gamma \supset [N/x]M{:}\beta$

(2) For any type-assignment η, for any assignment ρ such that $\eta \uparrow_\Gamma \rho$,

$$[\![\Gamma \supset N{:}\alpha]\!]\eta\rho\Downarrow \;\Rightarrow\; [\![\Gamma \supset [N/x]M{:}\beta]\!]\eta\rho \simeq [\![\Gamma, x{:}\alpha \supset M{:}\beta]\!]\eta\rho'$$

where: $\rho' \triangleq \rho[\, [\![\Gamma \supset N{:}\alpha]\!]\eta\rho \,/x\,]$.

Proof

Both assertions are proved by induction on the length of the typing.

(1) (*) $[N/x] * \equiv *$, and $\vdash\Gamma \supset *: 1$.

(asmp) Say: $y{:}\beta \in \Gamma \Rightarrow \Gamma, x{:}\alpha \supset y{:}\ \beta$. If $x \equiv y$ then $[N/x]y \equiv N$, and $\vdash\Gamma \supset N{:}\alpha$; otherwise $\vdash\Gamma \supset y{:}\beta$, by (asmp).

($\rightharpoonup$I) First we observe that a rule of exchange is derived in the sense that: $\vdash\Gamma, x{:}\alpha, y{:}\alpha' \supset M{:}\beta$ implies $\vdash\Gamma, y{:}\alpha', x{:}\alpha \supset M{:}\beta$ with a proof of the same length. Next, say: $\Gamma, x{:}\alpha, y{:}\beta \supset M{:}\ \beta' \Rightarrow \Gamma, x{:}\alpha \supset \lambda y{:}\beta.M{:}\ \beta \rightharpoonup \beta'$. Then we apply the ind. hyp. on $\Gamma, y{:}\beta, x{:}\alpha \supset M{:}\ \beta'$ to conclude: $\Gamma, y{:}\beta \supset [N/x]M{:}\ \beta'$. By ($\rightharpoonup$I) we conclude: $\vdash\Gamma \supset \lambda y{:}\beta.[N/x]M{:}\ \beta \rightharpoonup \beta'$.

($\rightharpoonup$E) Direct application of ind. hyp. .

($\forall$I) Say: $\Gamma, x{:}\alpha \supset M{:}\ \alpha \quad t \notin ftv(\Gamma, x{:}\alpha) \Rightarrow \Gamma, x{:}\alpha \supset (\lambda t.M){:}\ (\forall t.\alpha)$. By ind. hyp. $\vdash\Gamma \supset$

$[N/x]M$: α, and by ($\forall$I) we conclude: $\Gamma \supset (\lambda t.[N/x]M)$: $(\forall t.\alpha)$.

($\forall$E) Direct application of ind. hyp. .

(Y) Direct application of ind. hyp.

(2) Left to the reader. □

Canonical Forms are Defined *(Lemma 4.4)*

If $\vdash \varnothing \supset C{:}\alpha$ where C is a canonical form, then for any η and ρ, $[\![\varnothing \supset C{:}\alpha]\!]\eta\rho \Downarrow$.

Proof

We recall: C::= * | (λv: α. M) | (λtv. C). Hence we proceed by induction on the structure of C.

($C \equiv *$) Then $[\![\varnothing \supset *{:}1]\!]\eta\rho \Downarrow$.

($C \equiv \lambda x{:}\ \alpha.\ M$) Then $[\![\varnothing \supset \lambda x{:}\ \alpha.\ M{:}\ \alpha \rightharpoonup \beta]\!]\eta\rho \Downarrow$.

($C \equiv \lambda t.\ C$) Then, for any A, $[\![\varnothing \supset C{:}\ \alpha]\!]\eta[A/t]\rho \Downarrow$ by inductive hypothesis, and therefore $[\![\varnothing \supset \lambda t.C{:}\ \forall t.\alpha]\!]\eta\rho \Downarrow$. □

Invariance under Evaluation *(Lemma 4.5)*

If $\vdash M \mapsto C$ then, for any η and ρ, $[\![\varnothing \supset M{:}\alpha]\!]\eta\rho = [\![\varnothing \supset C{:}\alpha]\!]\eta\rho$.

Proof

By induction on the deduction of the evaluation judgment.

(*), ($\rightharpoonup$I) Trivial.

($\rightharpoonup$E) By applying the ind. hyp. and the term substitution lemma we have:

$[\![\varnothing \supset MN : \beta]\!]\eta\rho = [\![\varnothing \supset \lambda x{:}\alpha.M'{:}\ \alpha \rightharpoonup \beta]\!]\eta\rho\ [\![\varnothing \supset C'{:}\ \alpha]\!]\eta\rho =$

$[\![x{:}\alpha \supset M'{:}\ \beta]\!]\eta\rho[\ [\![\varnothing \supset C'{:}\ \alpha]\!]\eta\rho\ /x\ ] = [\![\varnothing \supset [C'/x]M'{:}\ \beta]\!]\eta\rho = [\![\varnothing \supset C{:}\ \beta]\!]\eta\rho$.

($\forall$I) By ind. hyp. .

($\forall$E) By applying the ind. hyp. and the type substitution lemma we have:

$[\![\varnothing \supset M\alpha{:}\ [\alpha/t]\beta]\!]\eta\rho = [\![\varnothing \supset (\lambda t.C){:}\ \forall t.\beta]\!]\eta\rho\ [\![\alpha]\!]\eta\rho =$

$[\![\varnothing \supset C{:}\ \beta]\!]\eta[[\![\alpha]\!]\eta/t]\rho = [\![\varnothing \supset [\alpha/t]C{:}\ [\alpha/t]\beta]\!]\eta\rho$.

(Y) By ind. hyp. and $[\![\varnothing \supset M(\lambda x{:}1.YM) : \alpha]\!]\eta\rho = [\![\varnothing \supset (YM) : \alpha]\!]\eta\rho$. □

Appendix B

In this section we build a realizability structure D such that $[\![\Omega]\!]^D\Downarrow$, where $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$. The construction is a straightforward adaptation of the $P\omega$ model construction, and of certain observations in Baeten&Boerboom[79] on the surprising effects that certain codings can have on the interpretation.

Consider the collection of subsets of natural numbers ordered by inclusion, say $P\omega$, as an object of the category **dcpo**. Suppose we are given two *bijective* codings: $<,>: \omega\times\omega\rightarrow\omega$, $e: \omega\rightarrow P_{fin}\omega$. We define $Graph_v$: $(P\omega\rightharpoonup P\omega)\rightharpoonup P\omega$, and Fun_v: $P\omega\rightharpoonup(P\omega\rightharpoonup P\omega)$ as follows:

$$Graph_v(f) = \{<n, m> \mid f(e_n)\Downarrow\ \wedge\ m\in f(e_n)\}$$
$$Fun_v(X) = \lambda Y.\ \text{if } Z\neq\varnothing \text{ then } Z \text{ else } \Uparrow,$$
$$\text{where: } Z = \{m \mid \exists n.(<n, m>\in X\ \wedge\ e_n\subseteq Y)\}$$

Verify the $\mathrm{Fun}_v \circ \mathrm{Graph}_v = \mathrm{id}_{(P\omega \to P\omega)}$. We set: $XY \equiv \mathrm{Fun}_v(X)(Y)$, and we denote with $P\omega^v$ the realizability structure resulting from this construction.

Next we expand the definition of $[\![\Omega]\!]^{P\omega^v}$.

$XX \cong \{m \mid \exists n.(\langle n, m\rangle \in X \wedge e_n \subseteq X)\}$ if not-empty, $\Uparrow$ o.w.

For $\omega \equiv \lambda x.xx$, $[\![\omega]\!]^{P\omega^v} = \mathrm{Graph}_v(\lambda X.\, XX) = \{\langle n, m\rangle \mid e_n e_n \Downarrow \wedge m \in e_n e_n\} = \{\langle n, m\rangle \mid m \in e_n e_n\} = \{\langle n, m\rangle \mid \exists x.(\langle x, m\rangle \in e_n \wedge e_x \subseteq e_n)\}$.

Hence $[\![\Omega]\!]^{P\omega^v} \cong \{z \mid \exists w.(\langle w, z\rangle \in [\![\omega]\!]^{P\omega^v} \wedge e_w \sqsubseteq [\![\omega]\!]^{P\omega^v})\}$ if not-empty, $\Uparrow$ o.w.

Let us now consider codings[2] such that:

$\langle 1, 0\rangle = 0$, and $e_1 = \{0\}$.

We claim: $0 \in [\![\Omega]\!]^{P\omega^v}$. Take w=1. Then:

(i) $\langle 1, 0\rangle \in [\![\omega]\!]^{P\omega^v}$ iff $\exists x.(\langle x, 0\rangle \in e_1 \wedge e_x \subseteq e_1)$.

But take x=1 and we have: $(\langle 1, 0\rangle = 0 \in e_1 \wedge e_1 \subseteq e_1)$.

(ii) $e_1 \sqsubseteq [\![\omega]\!]^{P\omega^v}$ iff $0 \in e_1 e_1$ iff $\exists n.(\langle n, 0\rangle \in e_1 \wedge e_n \subseteq e_1)$.

But take n=1 and we have: $(\langle 1, 0\rangle \in e_1 \wedge e_1 \subseteq e_1)$.

Conclusion, for such codings: $[\![\Omega]\!]^{P\omega^v} \Downarrow$. □

[2] Such codings are effective and can be easily built.

ISSN 0249-6399